Khushal kharade **EE779 ICA assignment** **10D070023**

Brief description of algorithm:

Preprocessing for ICA:

Centering:
This is done to center **x**, i.e. subtract its mean vector **m** = **E {x}** so as to make **x** a zero-mean variable. This implies that **s** is zero-mean as well, as can be seen by taking expectations on both sides of **x = As.**
After estimating the mixing matrix **A** with centered data, we can complete the estimation by adding the
mean vector of **s** back to the centered estimates of **s**. The mean vector of **s** is given by $\mathbf{A}^{-1}\mathbf{m}$, where **m** is the mean that was subtracted in the preprocessing.

Whitening:
In FastICA the whitening matrix performs whitening and the reduction of dimension. Before the application of the ICA algorithm (and after centering), the observed vector **x** is transformed linearly to obtain a new vector $\tilde{\mathbf{x}}$ which is white, i.e. its components are uncorrelated and their variances equal unity.
Whitening transforms the mixing matrix into a new one, $\tilde{\mathbf{A}}$, using EVD covariance matrix of **x.** whitening reduces the number of parameters to be estimated. Instead of having to estimate the n^2 parameters that are the elements of the original matrix **A**, we only need to estimate the new, orthogonal mixing matrix **A**. An orthogonal matrix contains $n(n - 1)/2$ degrees of freedom. For example, in two dimensions, an orthogonal transformation is determined by a single angle parameter. In larger dimensions, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix. So, the computational complexity has been reduced significantly.

Here, $\mathbf{g = kurt(y) = E\ \{y^4\} - 3(E\ \{y^2\})^2}$.

The algo starts from some weight vector **w**, compute the direction in which the kurtosis of $\mathbf{y} = \mathbf{w}^T\mathbf{x}$ is growing most strongly (if kurtosis is positive) or decreasing most strongly (if kurtosis is negative) based on the available sample **x**(1),..., **x**(T) of mixture vector x, and use the following method for finding a new vector w.

The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of $\mathbf{w}^T\mathbf{x}$.

For one-unit,

The basic form of the FastICA algorithm is as follows:

1. Choose an initial (e.g. random) weight vector **w**.

2. Let $\mathbf{w+} = \mathbf{E}\ \{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \mathbf{E}\ \{\mathbf{g0}(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$

3. Let $\mathbf{w} = \mathbf{w+}/\|\mathbf{w+}\|$

4. If not converged, go back to 2.

Here, convergence means that the old and new values of w point in the same direction, i.e. their dot-product is (almost) equal to 1. It is not necessary that the vector converges to a single point, since **w** and −**w** define the same direction.

For several-units,

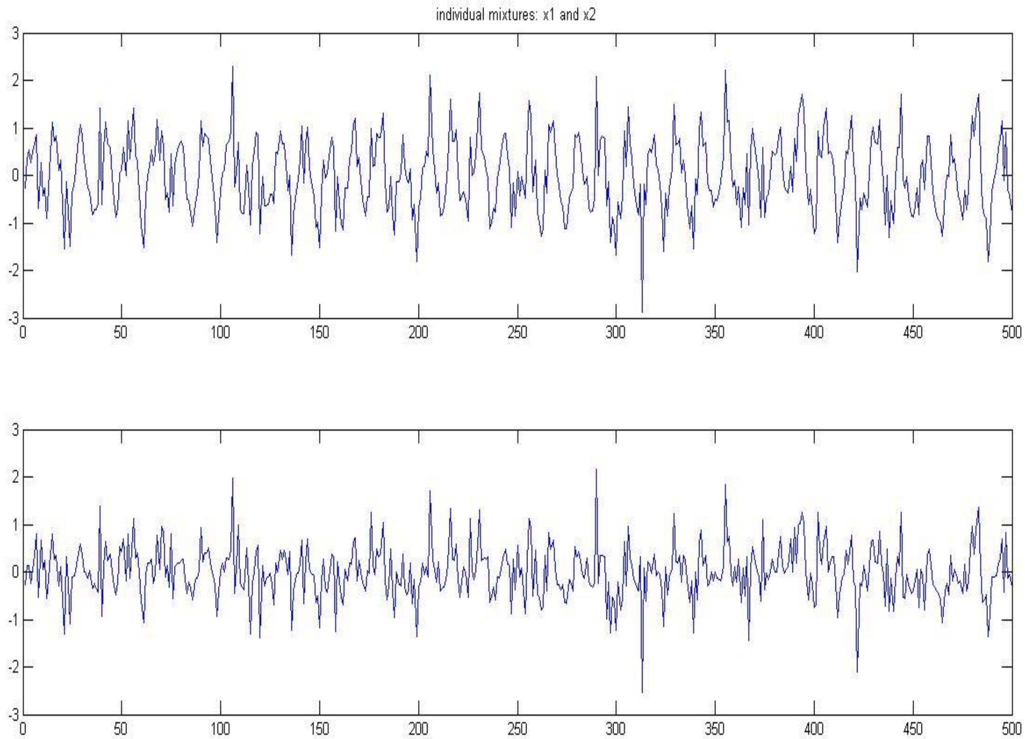
To prevent different vectors from converging to the same maxima, the outputs are decorrelated $\mathbf{w_1}^T\mathbf{x}$,..., $\mathbf{w_n}^T\mathbf{x}$ after every iteration. This is achieved by a deflation scheme based on a Gram-Schmidt-like decorrelation. i.e. estimating the independent components one by one. After estimating **p** independent components, or **p** vectors $\mathbf{w_1,..., w_p}$, run the one-unit fixed-point algorithm for $\mathbf{w_p+1}$, and after every iteration step subtract from $\mathbf{w_p+1}$ the "projections" $\mathbf{w_{p+1}}^T\mathbf{w_j w_j}$, j = 1,..., p of the previously estimated p vectors, and then $\mathbf{w_{p+1}}$ is renormalize

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}}^T\mathbf{w_j w_j}$$

Results:

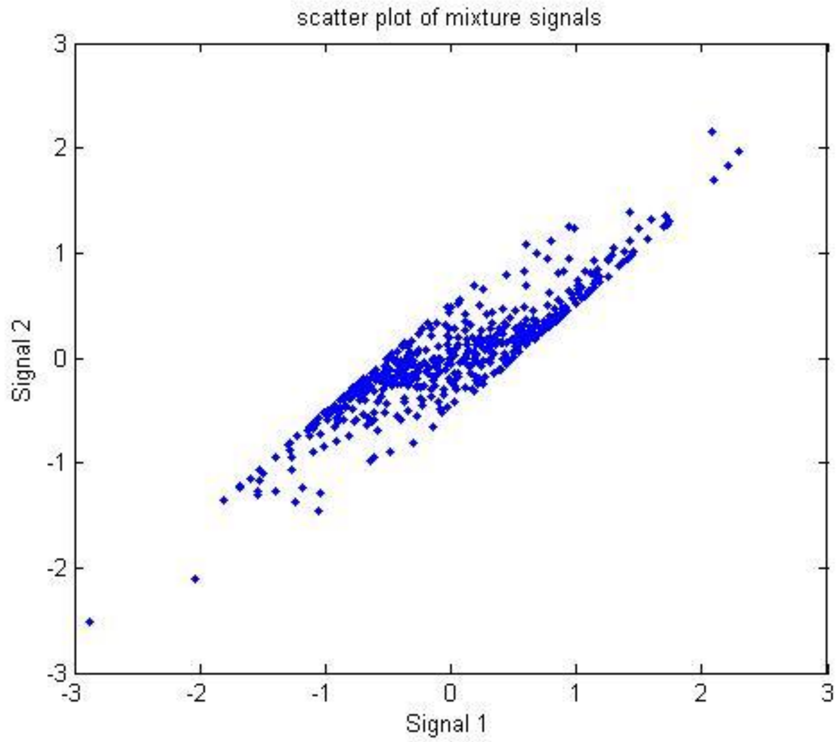
Q.4 Synthetic signal

a) *Observed Mixture of source signals*. x1 and x2



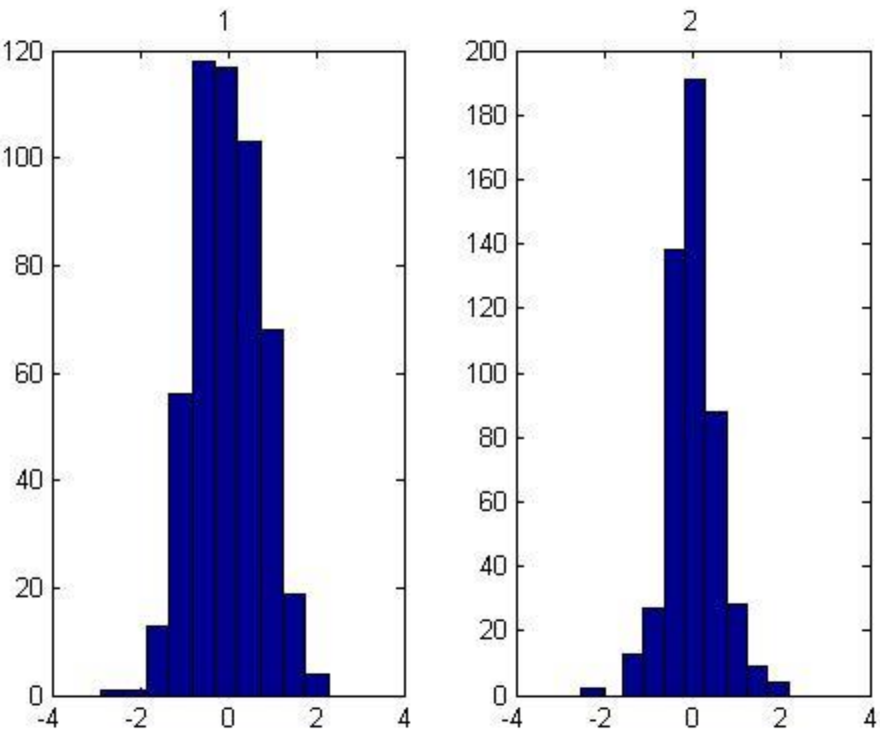
From the mixed signals, it seems that one of the signals is very much a random noise.

Scatter plot of mixture signals

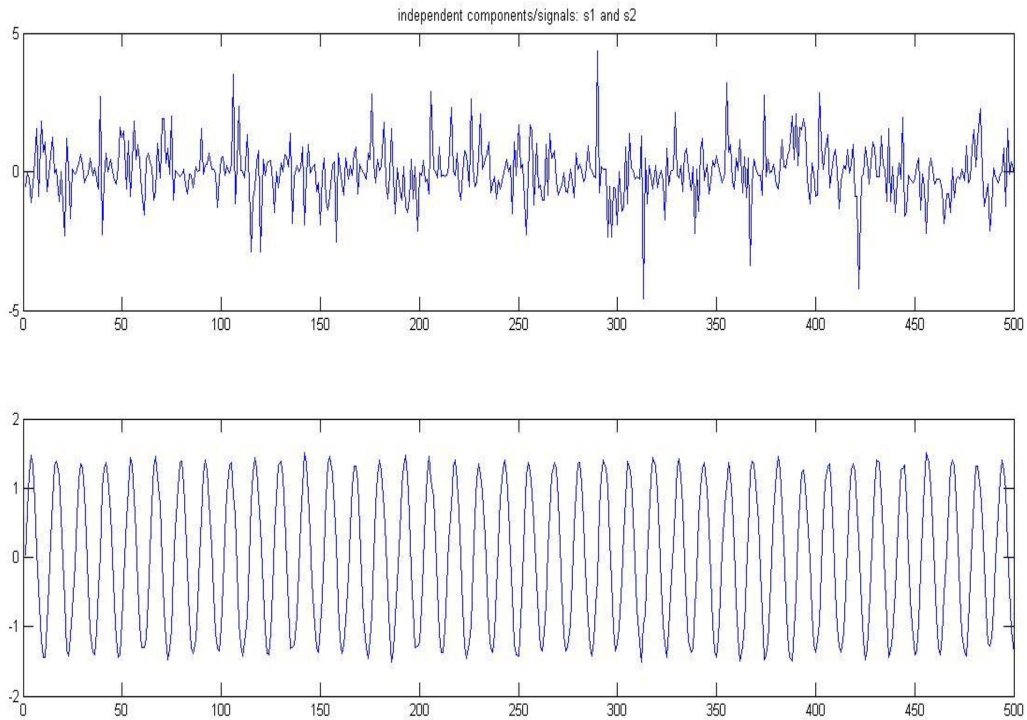


No symmetry in Scatter plot implies that mixed signals are not independent, so we should be able to separate two independent components out of the mixture.

__Histograms of x1 and x2__

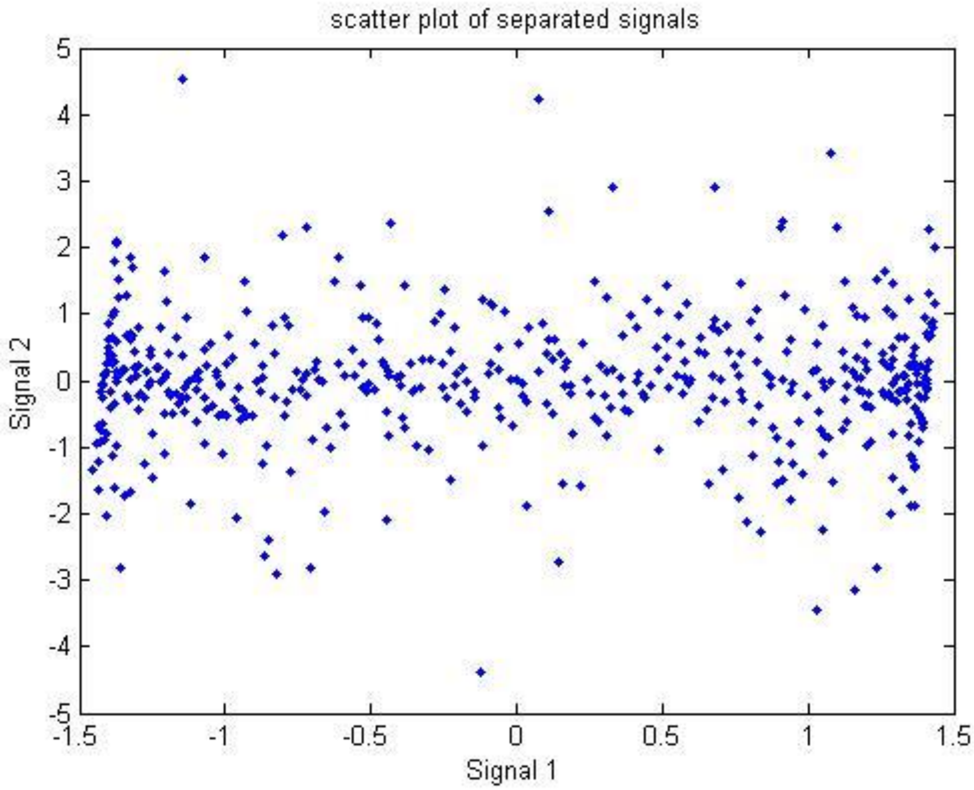


b) *Independent components/separated signals*: s1 and s2



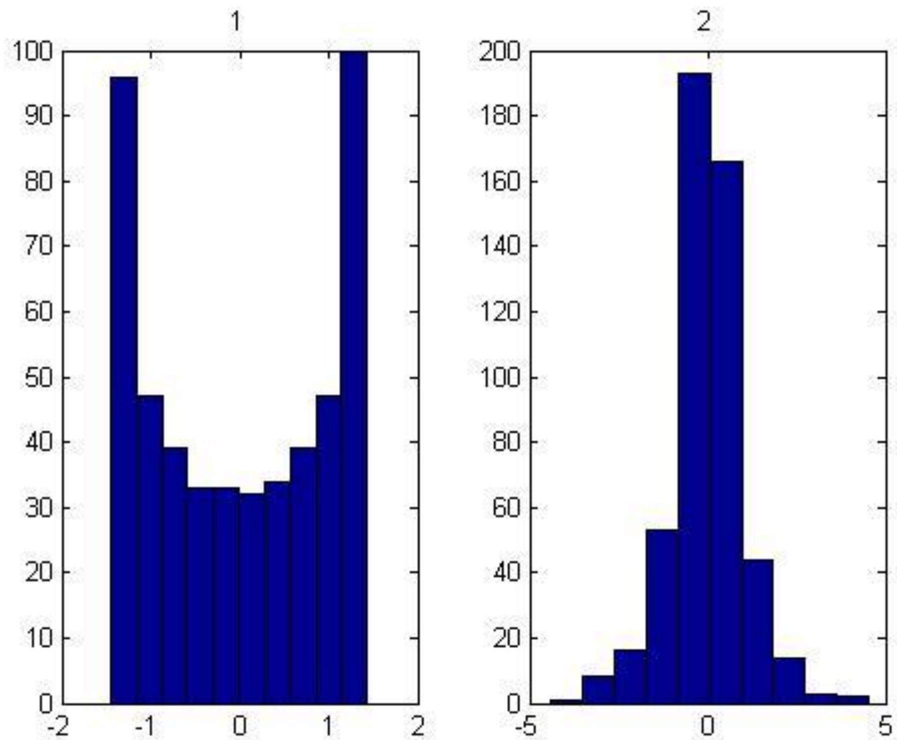
The signals seem to be accurately separated.

<u>Scatter plot of independent component signals</u>



Now, this scatter plot looks symmetric, which concludes that the separated signals are independent.

Histogram of s1 and s2:

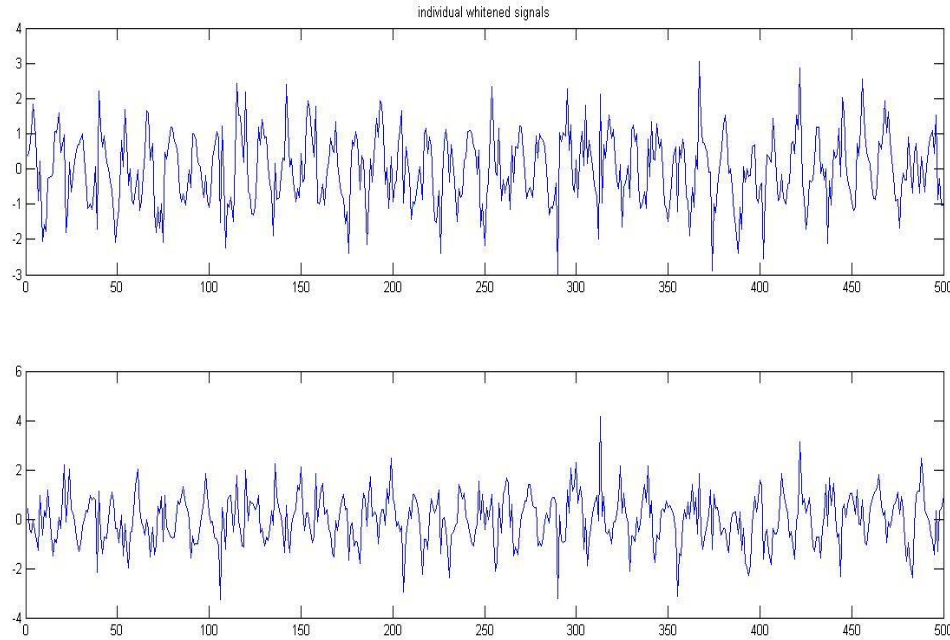


We can confer from the histogram that, signal 1 is non-gaussian while signal 2 looks like super-Gaussian.

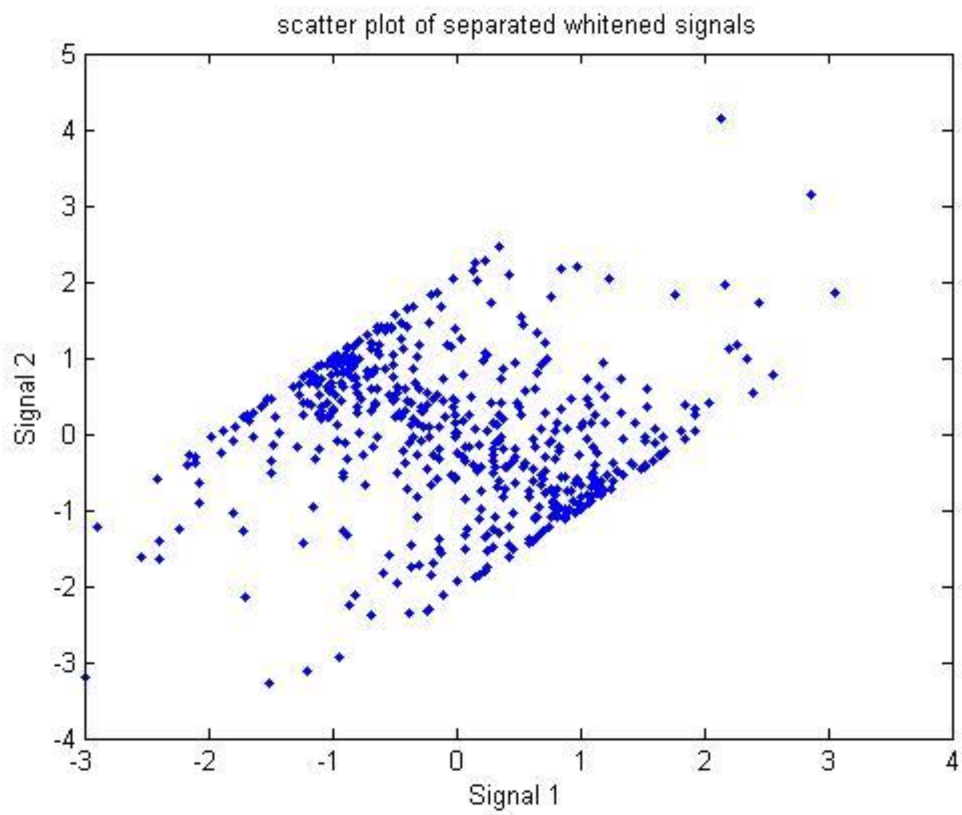
c) A = mixing matrix

0.578561506063927	-0.490261688054778
0.232564299027186	-0.498427095251282

d) Only with ‘white’.
Individual signals with only whitening

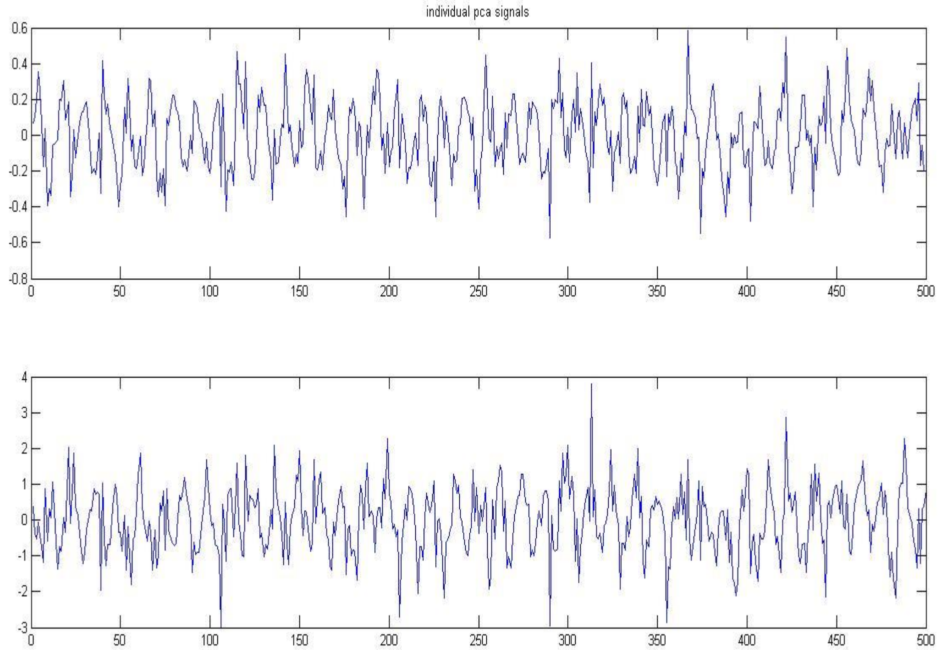


Scatter plot of signals with only whitening

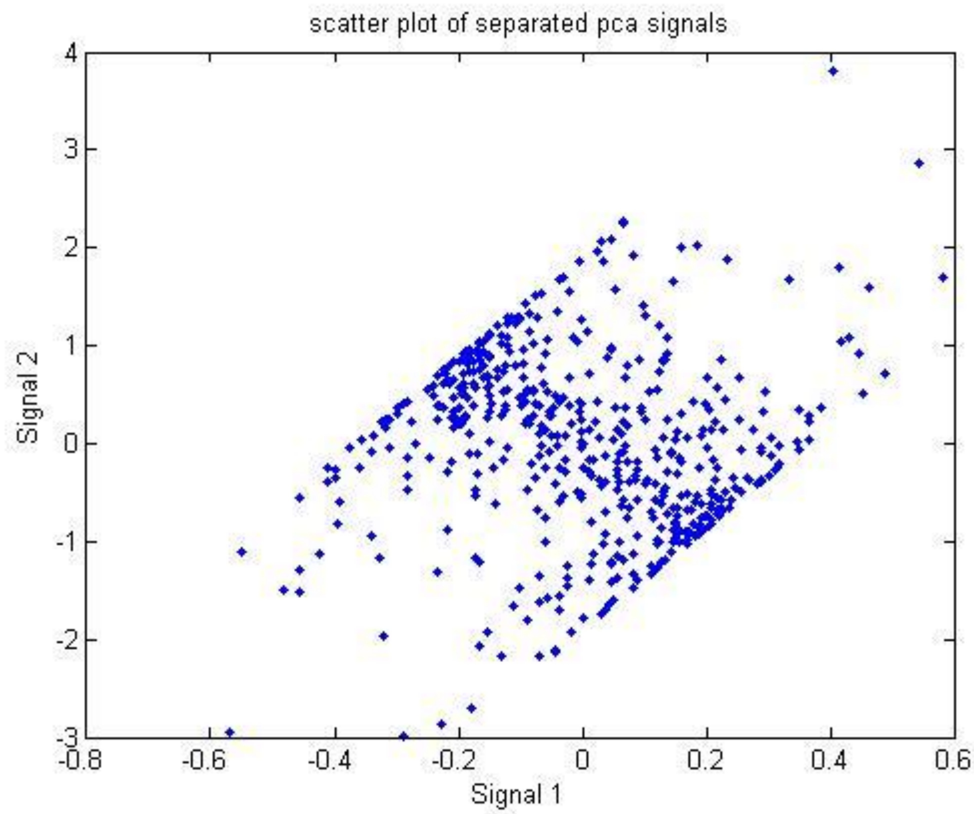


Only with pca

Individual signals with only pca

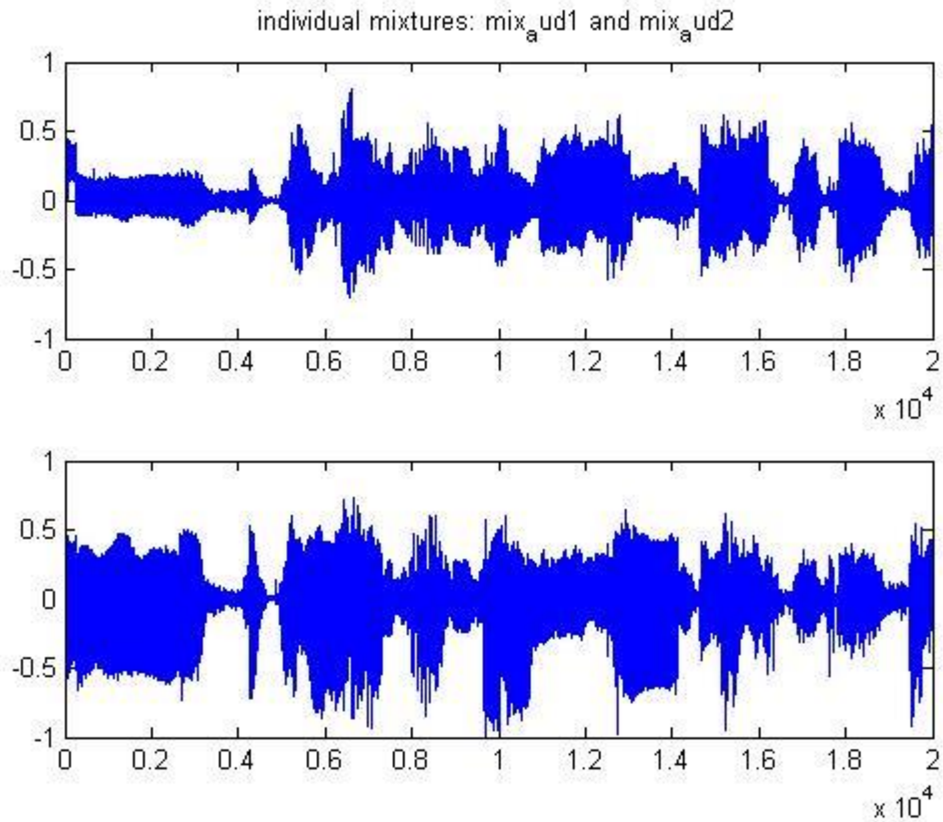


Scatter plot of signals only with pca

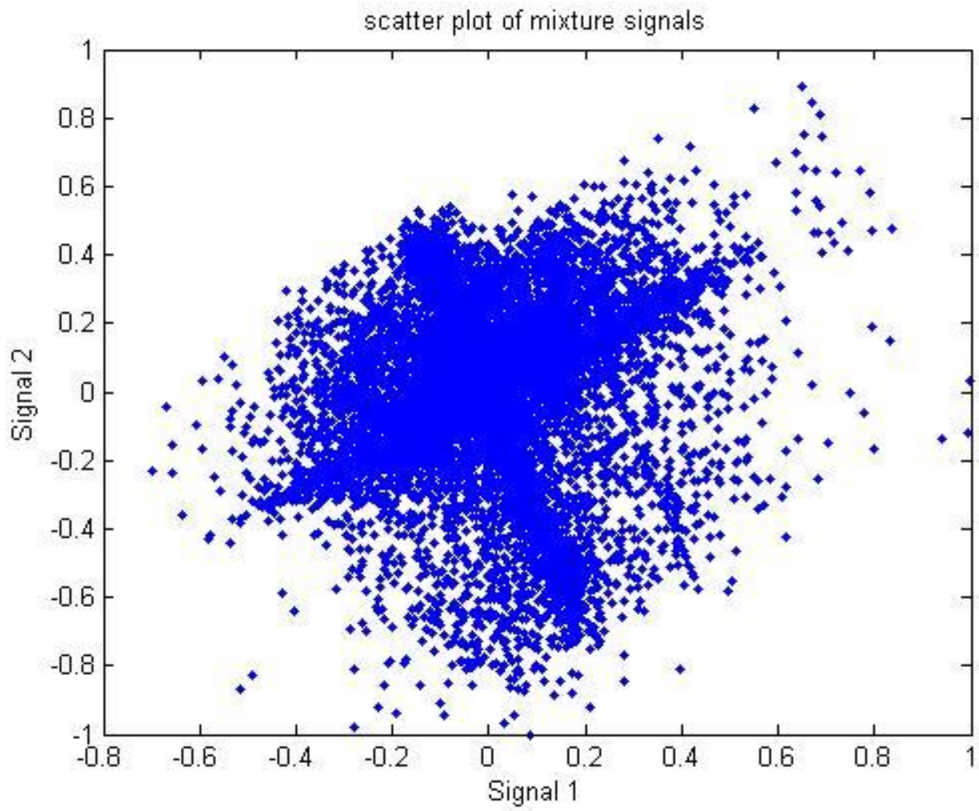


Q,5 Speech mixture signals

a. *Mixtures of audio signals*: mix_aud1 and mix_aud2

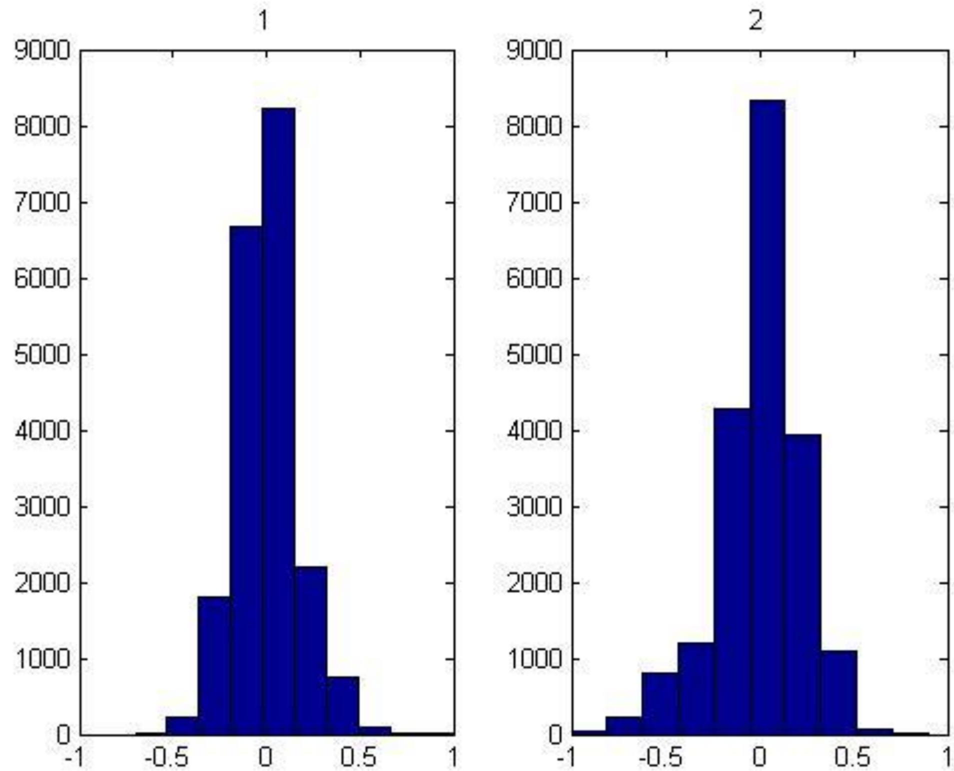


Scatter plot of mixture signals

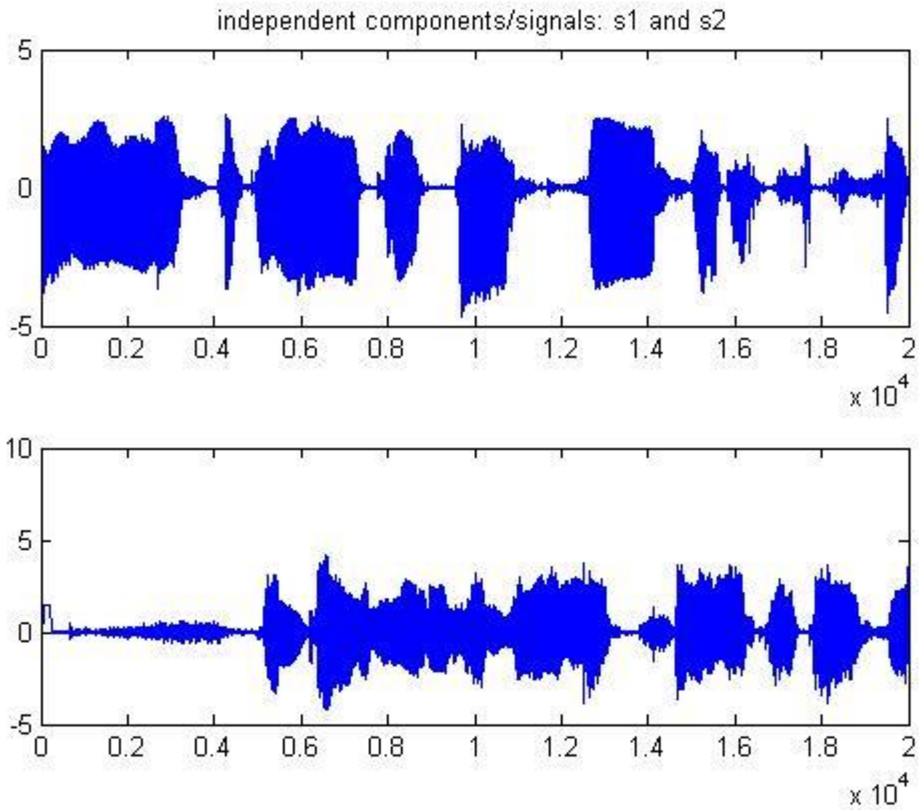


Here, also, scatter plot is not symmetric along horizontal (signal1) or vertical (singal2) axis. It implies that mixed audio signals are not independent, so we should be able to separate two independent components out of this mixture.

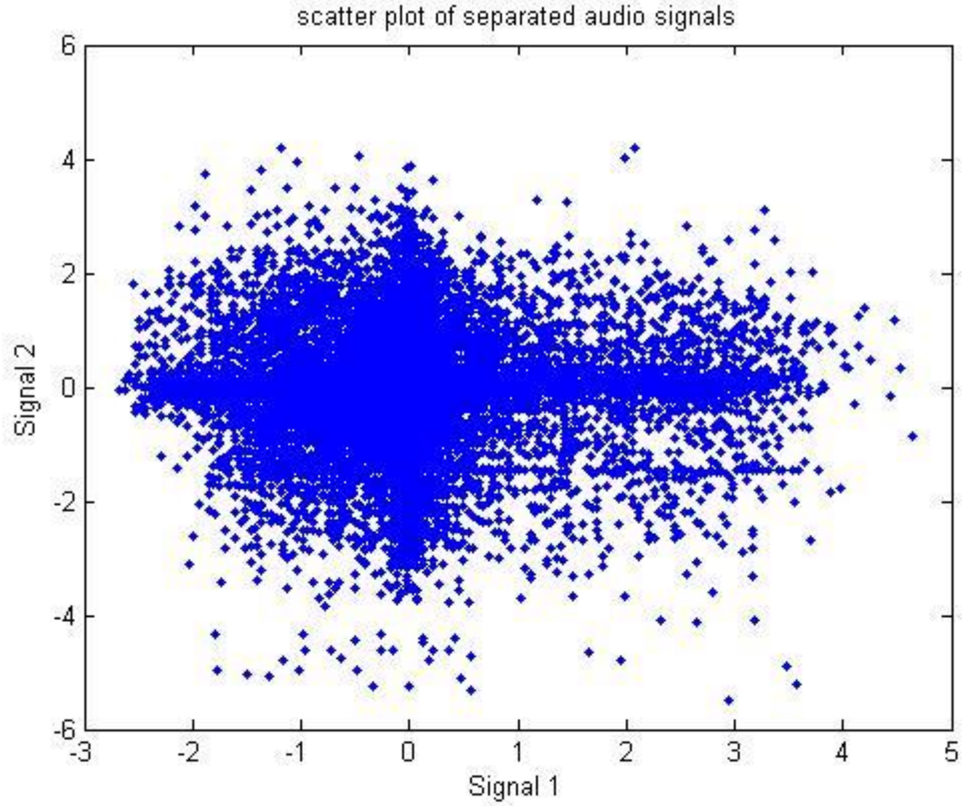
Histogram of mixture of audio signals



b. *Independent components/separated audio signals*: ica_audsig1 and ica_audsig2

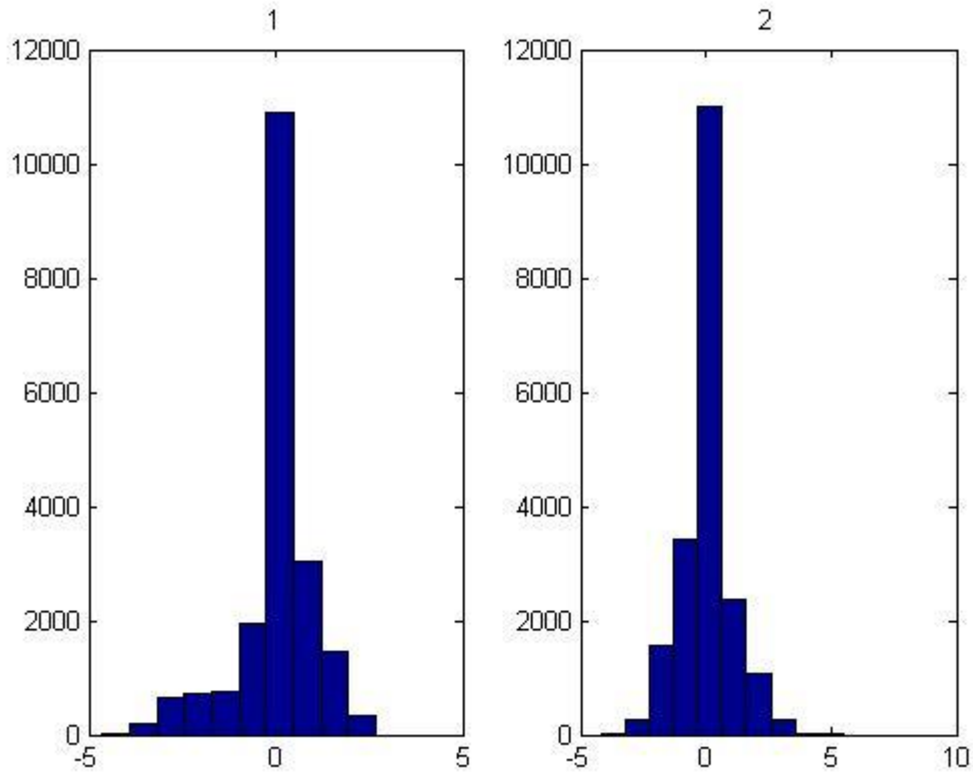


<u>*Scatter plot of independent audio signals*</u>: s1 vs s2



The above symmetric plot shows that components in the mixed audio signals are surely independent.

Histogram of separated/independent audio signals



From histogram, we can say that the separated audio signals have super-gaussian distribution

c. A = mixing matrix

-0.0576725672050978	-0.150861131505366
0.194678724767955	-0.110194394614214

The separated signals are quite audible.

One of the signals is by a male and it goes something like this: **The guardians of the electronics stock…**

The other signal is by a female and goes like this: **All list of the American fast-food that have succ…**

Optional comments:

The edges of the parallelogram in the scatter plot are in the directions of the columns of A.

The fundamental restriction in ICA is that the independent components must be non-gaussian for ICA to be possible. So, if there is Gaussian noise present in the mixture we will not be able separate that out if the original source signals are also Gaussian.

We can use the reduction of dimensionality in case we want less independent source signals. But if the number of mixtures is less than the number of source signals, then we can't use the above algorithm to estimate the independent components accurately.

Properties of FastICA algorithm

- The convergence is cubic (or at least quadratic), under the assumption of the ICA data model.
- The algorithm finds directly independent components of (practically) any non-Gaussian distribution using any nonlinearity g.
- The performance of the method can be optimized by choosing a suitable nonlinearity g.
- The independent components can be estimated one by one, which is roughly equivalent to doing projection pursuit.
- The FastICA has most of the advantages of neural algorithms: It is parallel, distributed, computationally simple, and requires little memory space.